Papis du Web
Épisode #6

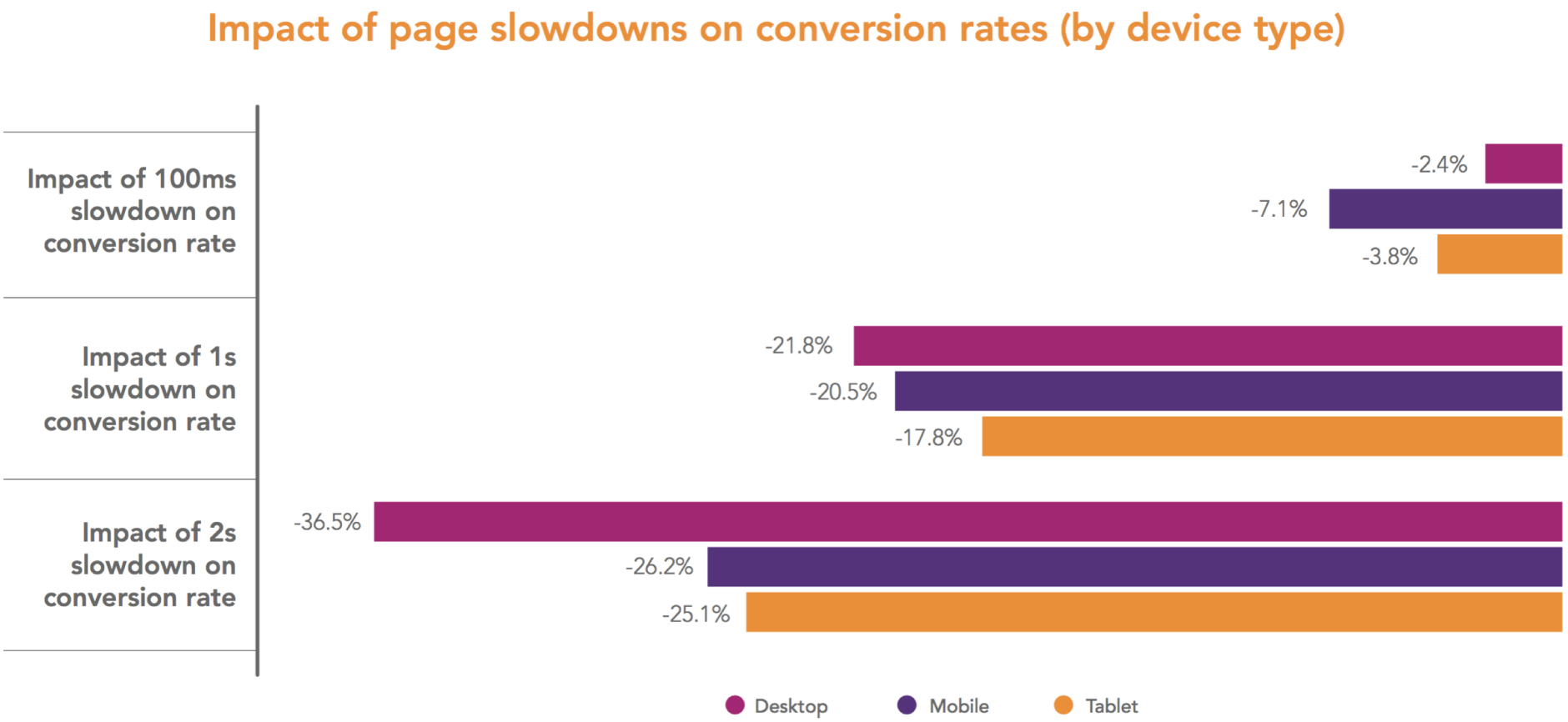
Performance Web

CLEVER AGE

La performance

un élément clef de l'UX

Une mauvaise performance fait fuire



Source : https://www.soasta.com/wp-content/uploads/2017/04/State-of-Online-Retail-Performance-Spring-2017.pdf

Une mauvaise performance fait fuire

As page load time goes from:

1s to 3s the probability of bounce **increases 32%**

1s to 5s the probability of bounce **increases 90%**

1s to 6s the probability of bounce **increases 106%**

1s to 10s the probability of bounce **increases 123%**

Google

think with Google — février 2018

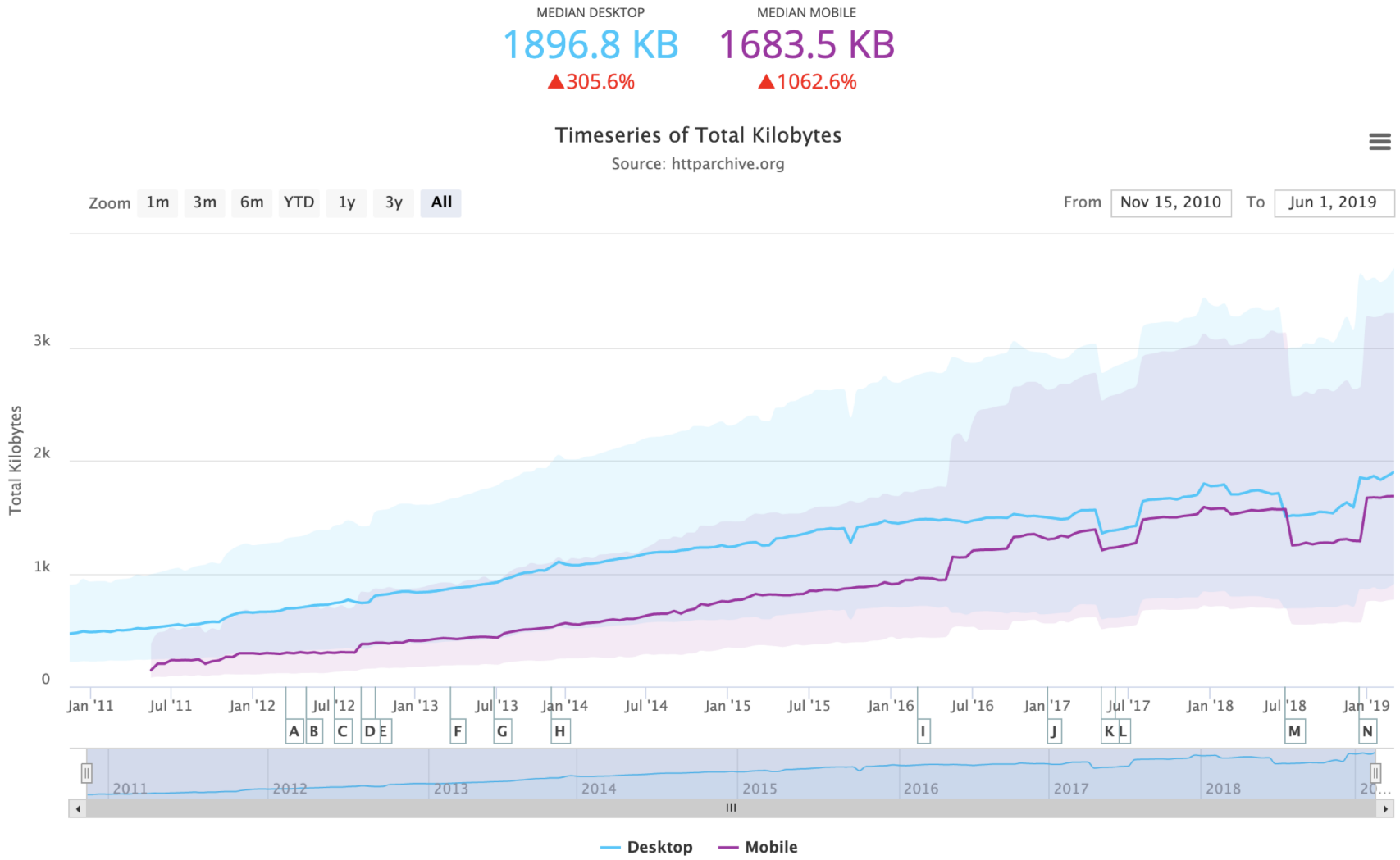
Source : https://www.thinkwithgoogle.com/marketing-resources/data-measurement/mobile-page-speed-new-industry-benchmarks/

Avec un site plus rapide…

- La satisfaction des utilisateur est améliorée
- Ils reviennent plus volontiers
- Ils regardent plus de pages à chaque visite
- Le taux de conversion et le CA augmentent
- Des économies d'infrastructure sont possibles
- Un meilleur positionnement chez Google

Pourquoi les sites sont-ils lents ?

Ils sont de plus en plus lourds



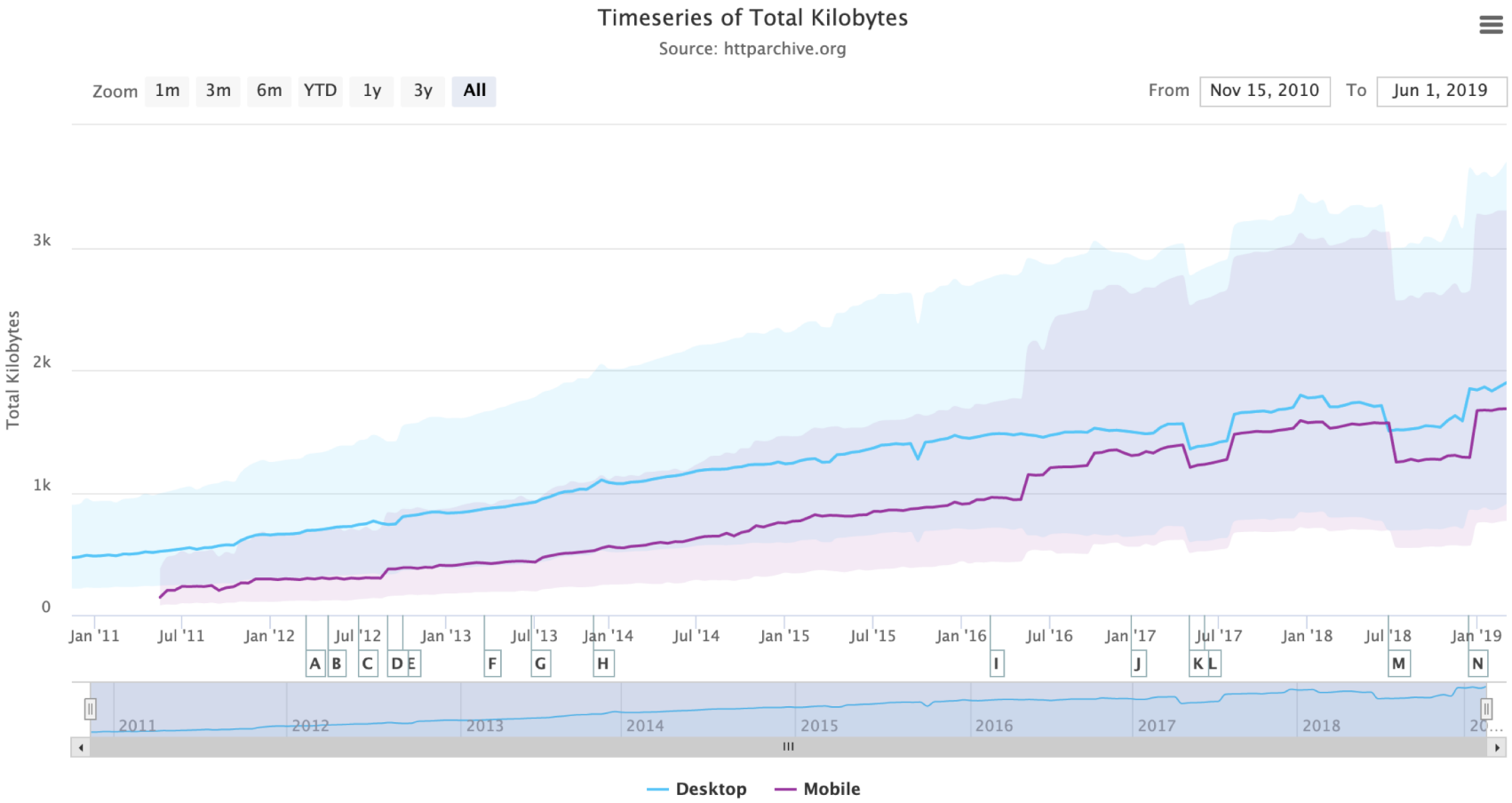
Source : https://httparchive.org/reports/state-of-the-web

Ils sont de plus en plus lourds

MEDIAN DESKTOP
1896.8 KB
▲305.6%

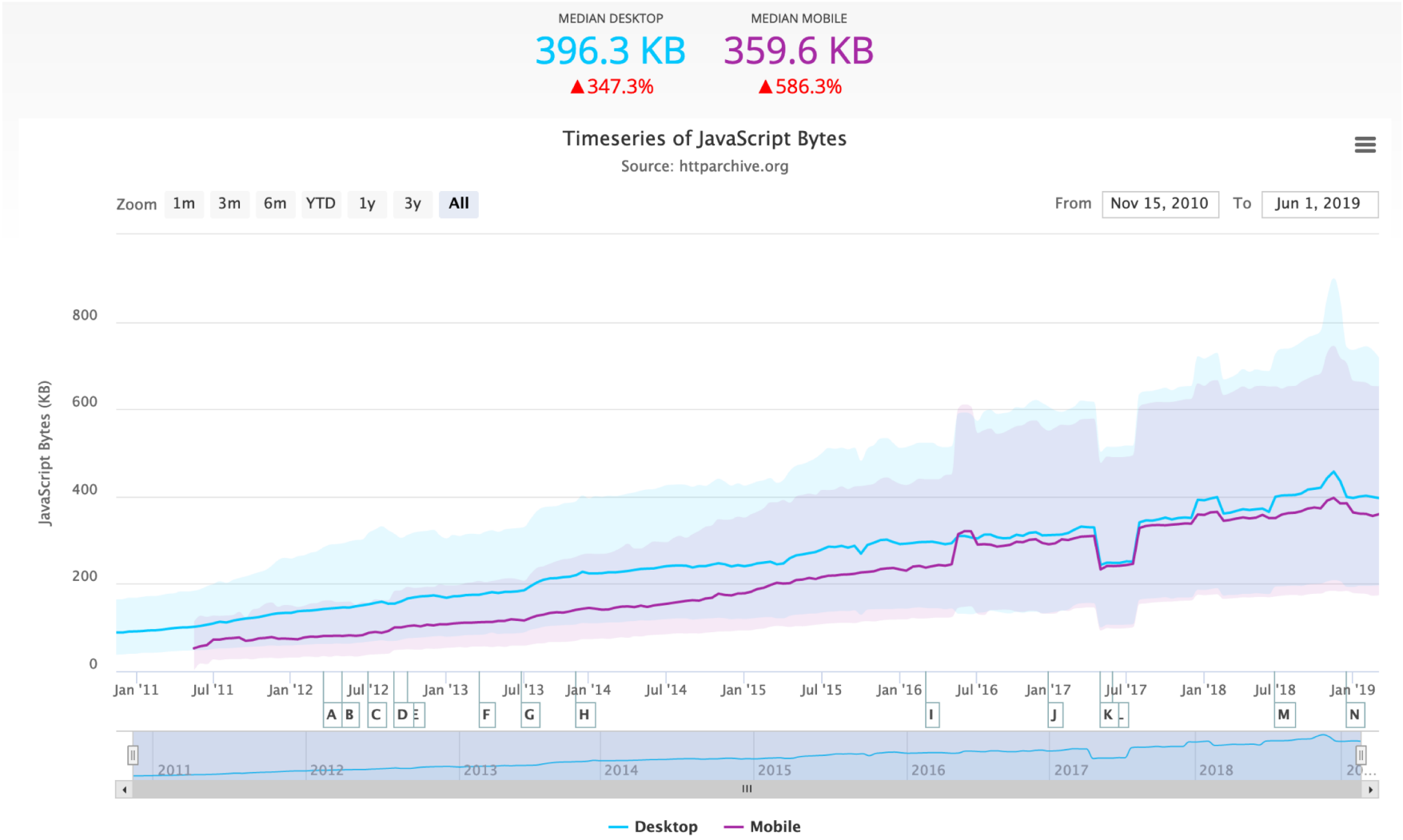
MEDIAN MOBILE
1683.5 KB
▲1062.6%

+1000 % 😱

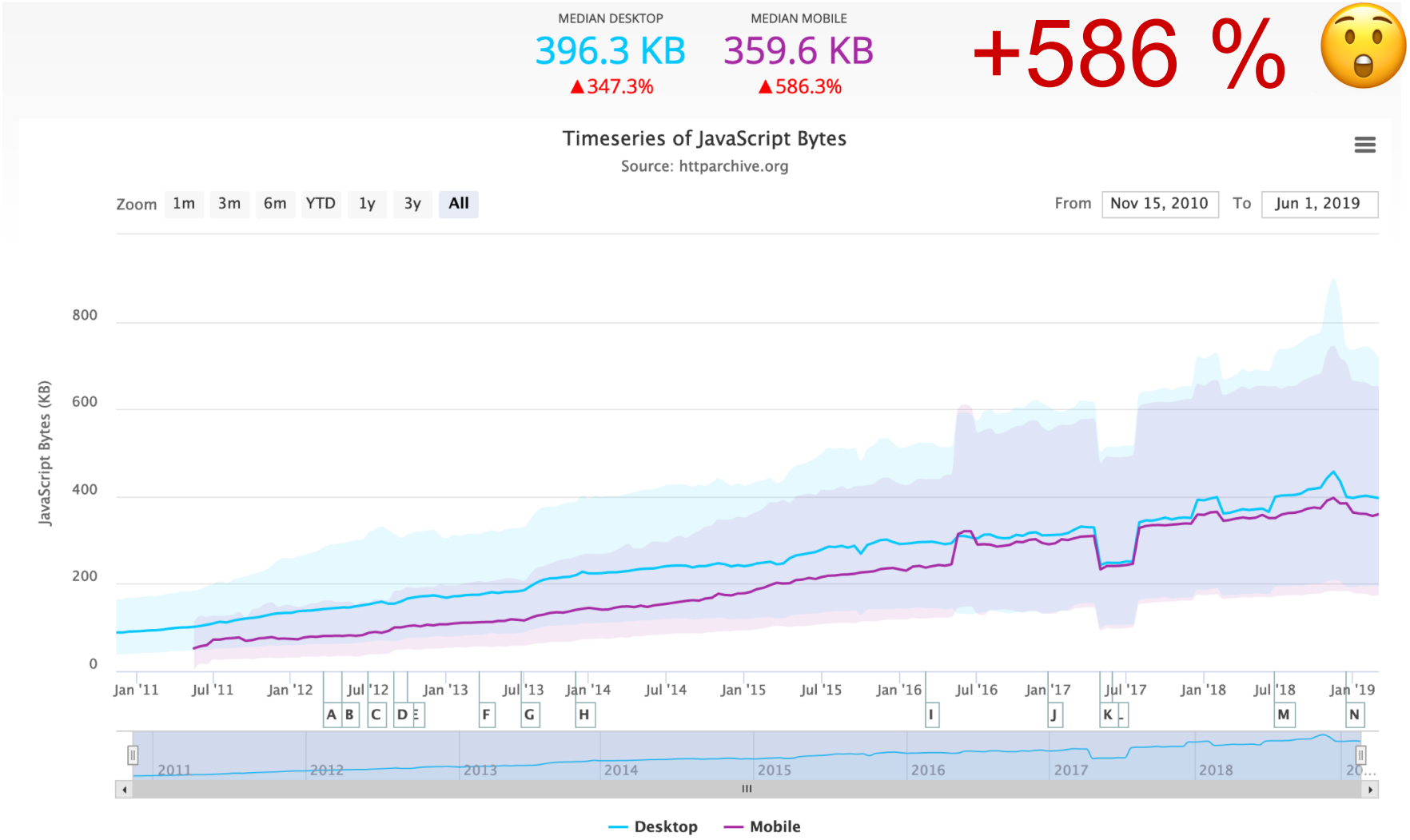


Source : https://httparchive.org/reports/state-of-the-web

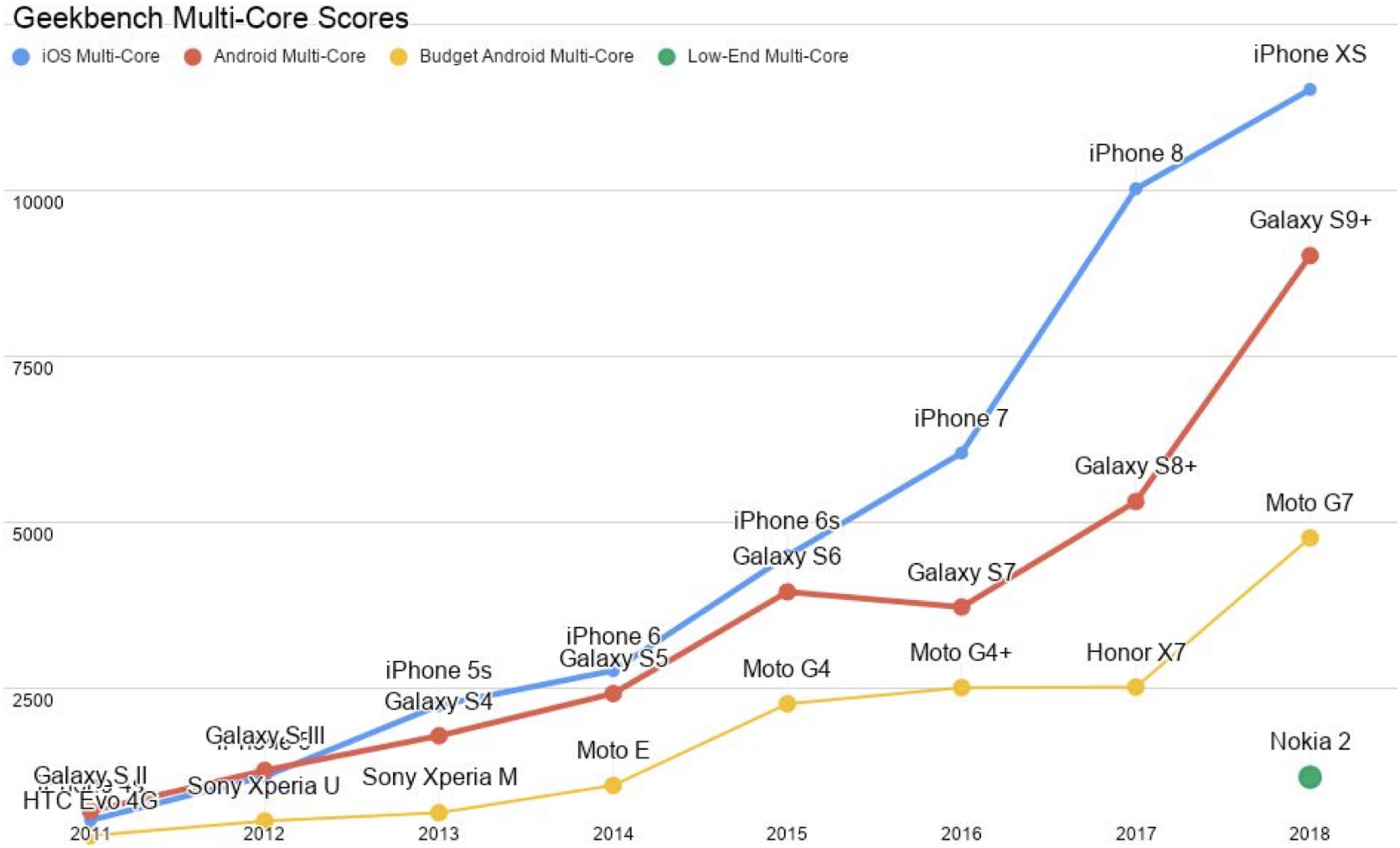
Ils sont saturés de JavaScript



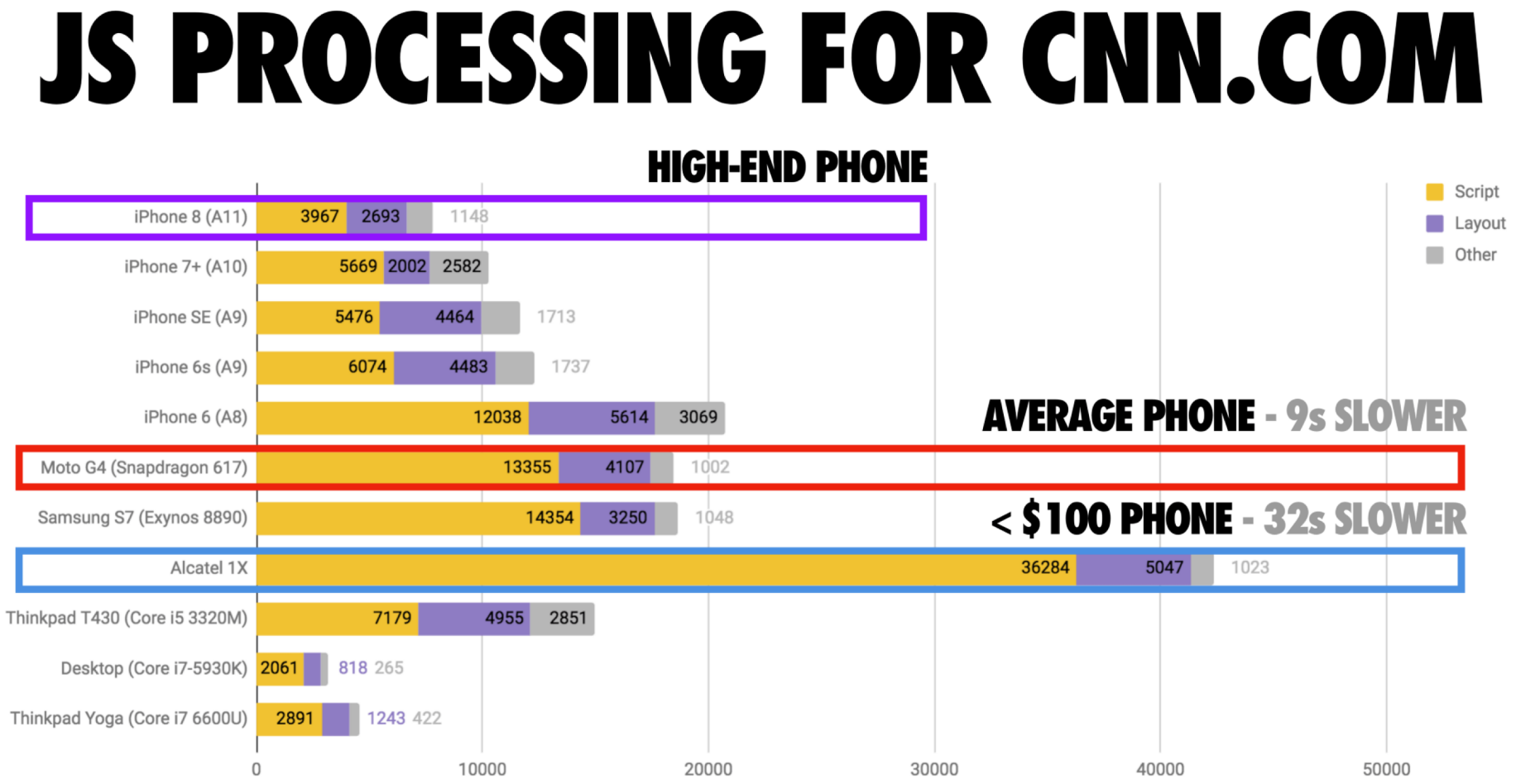
Ils sont saturés de JavaScript



Le JavaScript est la ressource la plus impactante

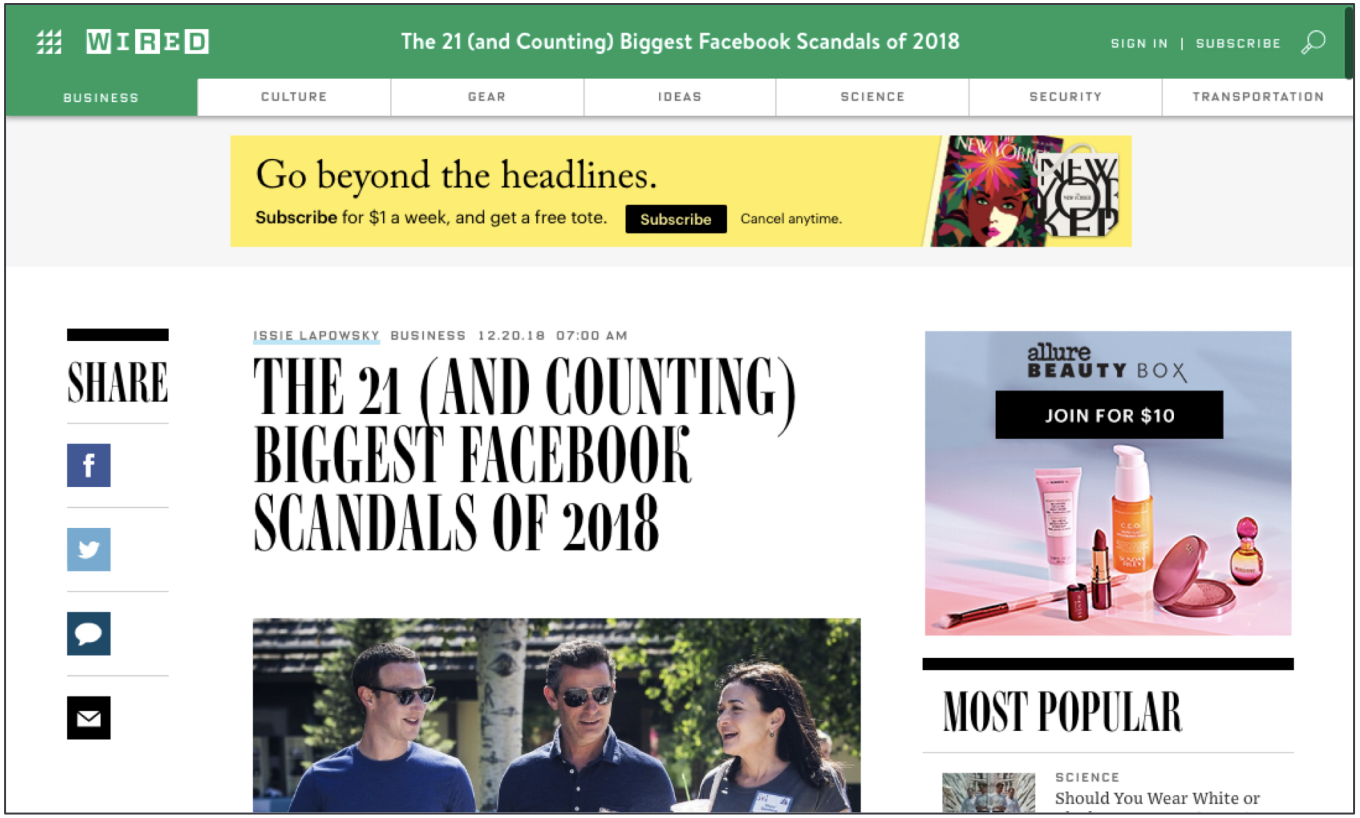


Le JavaScript est la ressource la plus



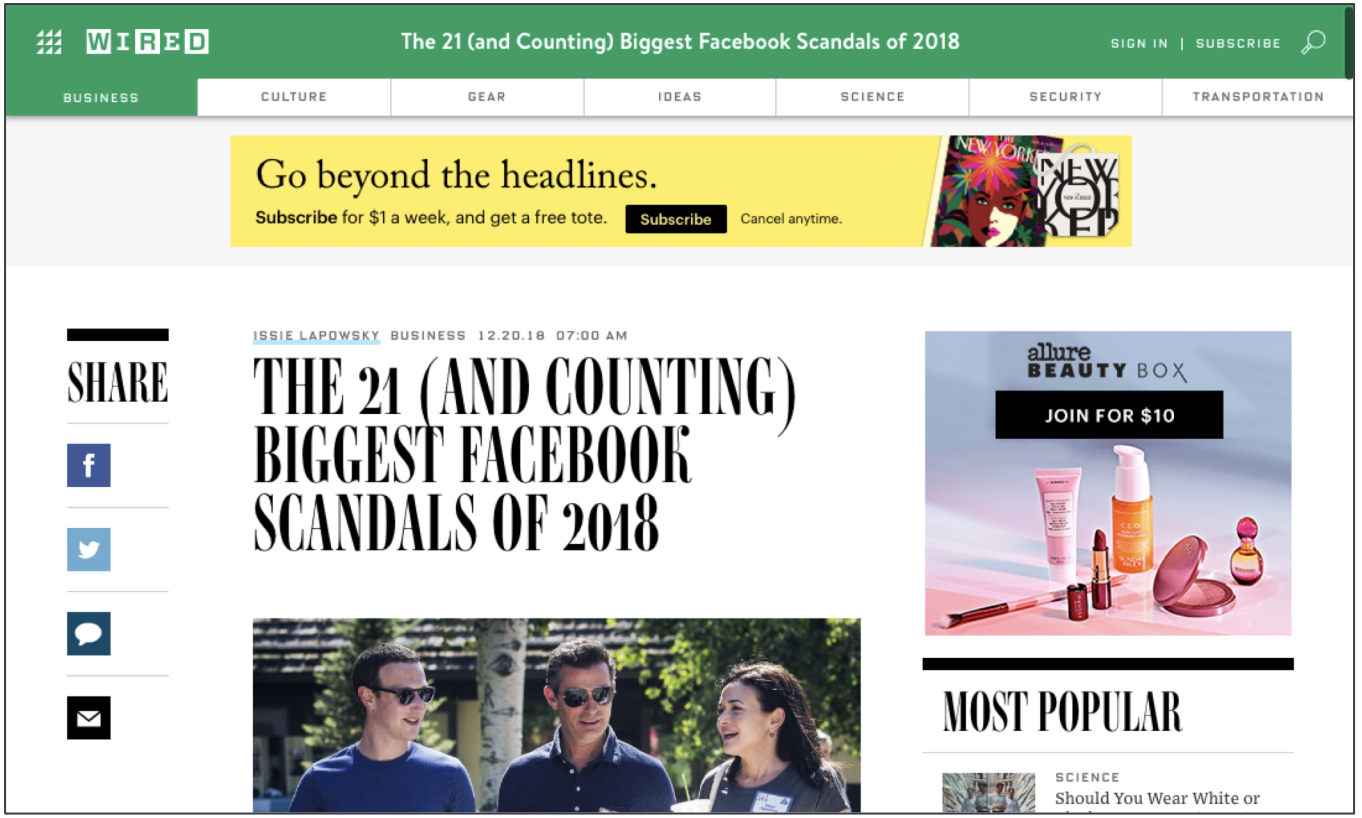
Source : https://medium.com/@addyosmani/the-cost-of-javascript-in-2018-7d8950fbb5d4

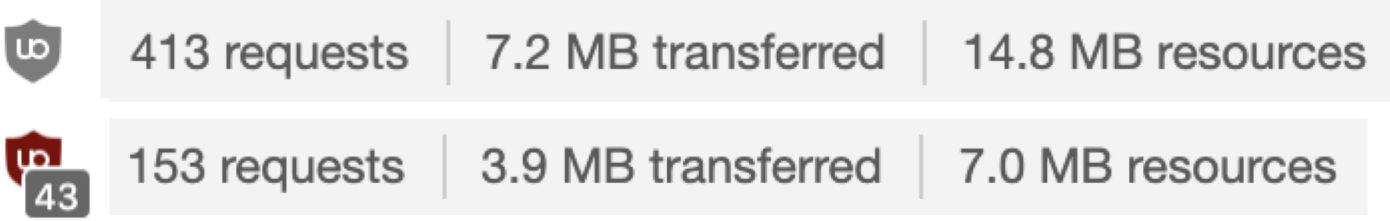
Les services tiers nous plombent



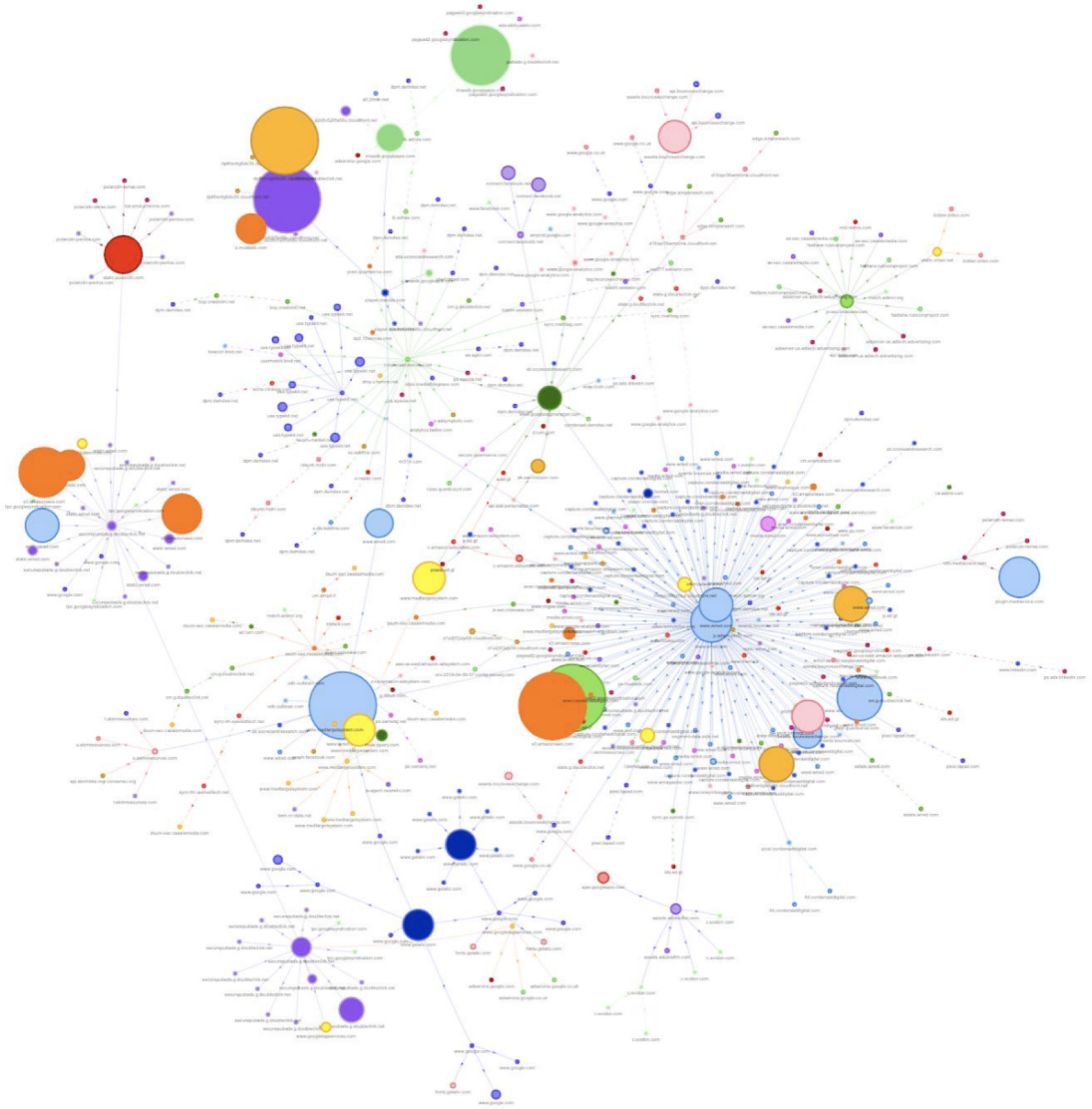
413 requests | 7.2 MB transferred | 14.8 MB resources

Les services tiers nous plombent





Feu d’artifice sur Request Map : mauvais signe



208 Cookies

- 54 first-party
- 154 third-party

533 third-party requests

170 unique hosts

Source : https://requestmap.webperf.tools/

Mais alors, comment faire ?

1. Mesurer

Mesurer en local

Browser devtools

Lighthouse

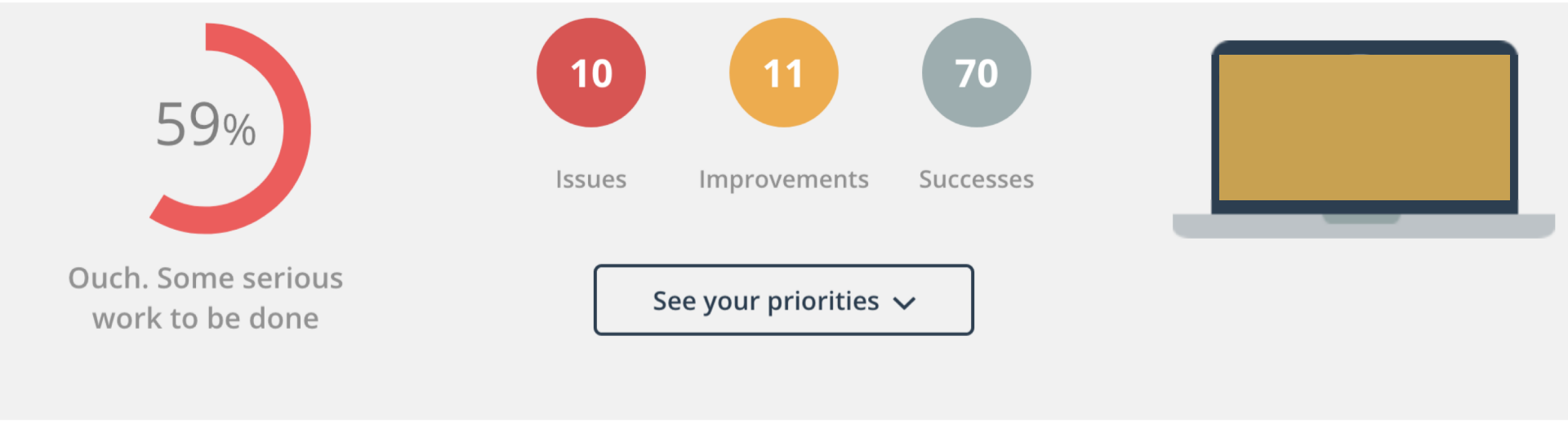
Etc.

Mesurer ponctuellement à distance

Outils de mesure synthétique

Etc.

Dareboost


59%
Ouch. Some serious work to be done
10
Issues
11
Improvements
70
Successes
See your priorities
SIMULATED VISITOR:
Chrome
Paris
8.0/1.5Mbps (Latency: 50 ms)
Edit
Memorize
Requests
93
Weight
5.6
Mo
HTML
CSS
Scripts
Images
Others
First Byte
0.59
sec
Start Render
1.50
sec
Fully loaded
7.33
sec

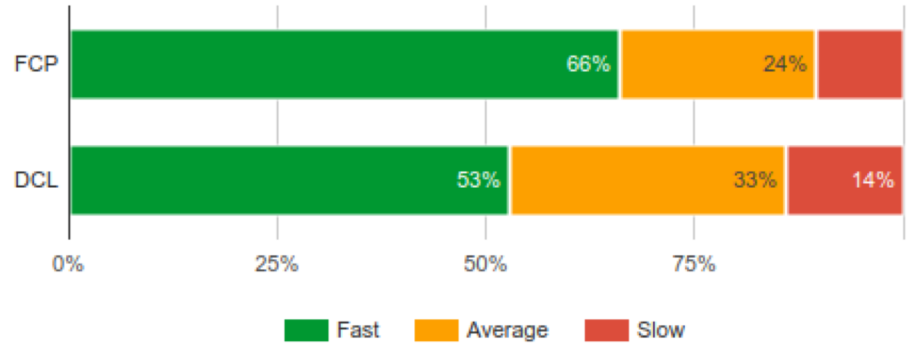
Google PageSpeed Insights

Mobile | Ordinateur

Vitesse	Optimisation
Fast	Good
1.2s FCP **2.0s** DCL	**93** / 100

Data from the Chrome User Experience report indicates this page's median FCP (1.2s) and DCL (2.0s) ranks it in the fastest third of all pages. This page has a good level of optimization because few of its resources are render-blocking. En savoir plus

Distributions des chargements de page



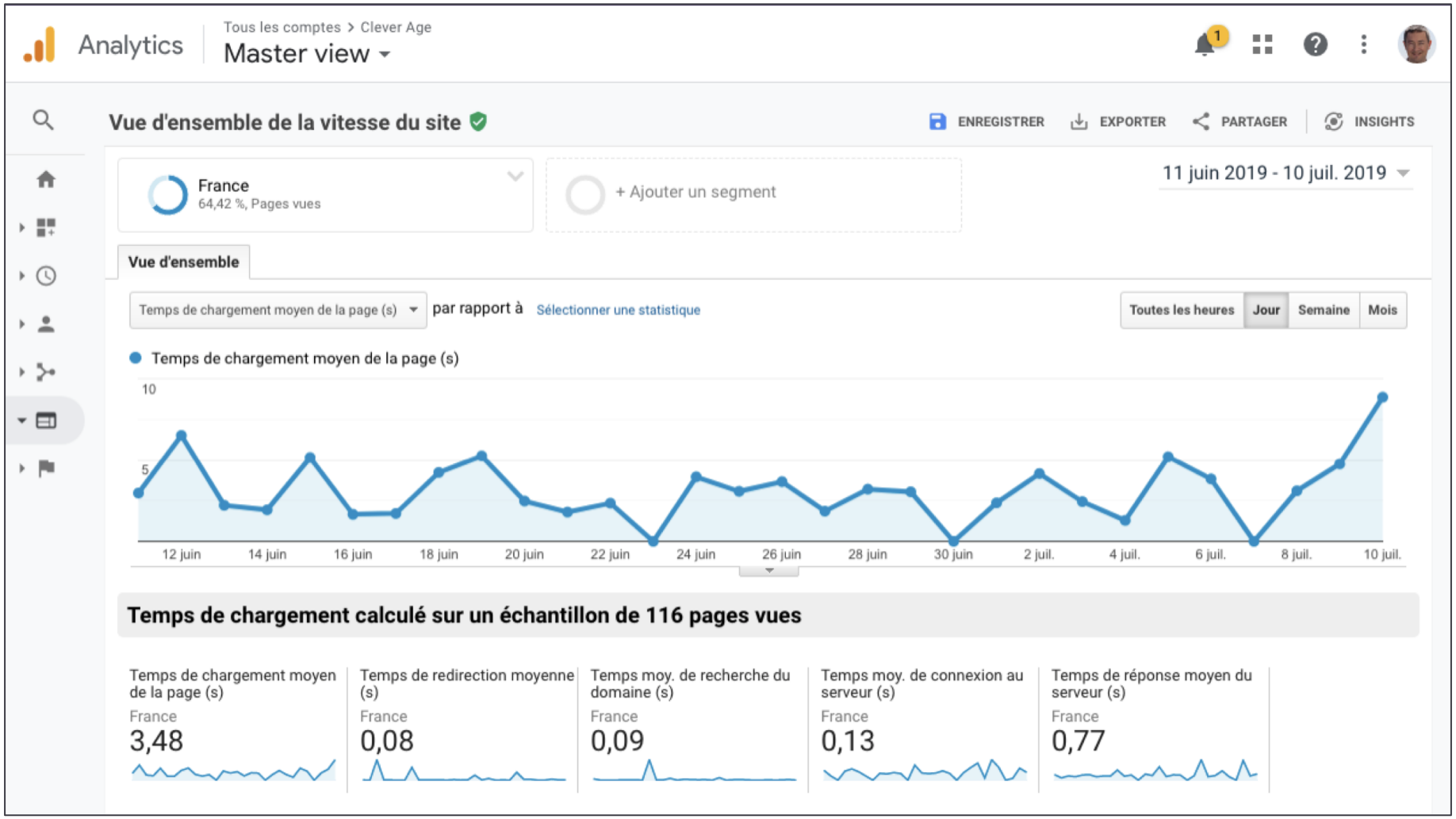
Mesurer en conditions d'utilisation réelle

Real User Monitoring

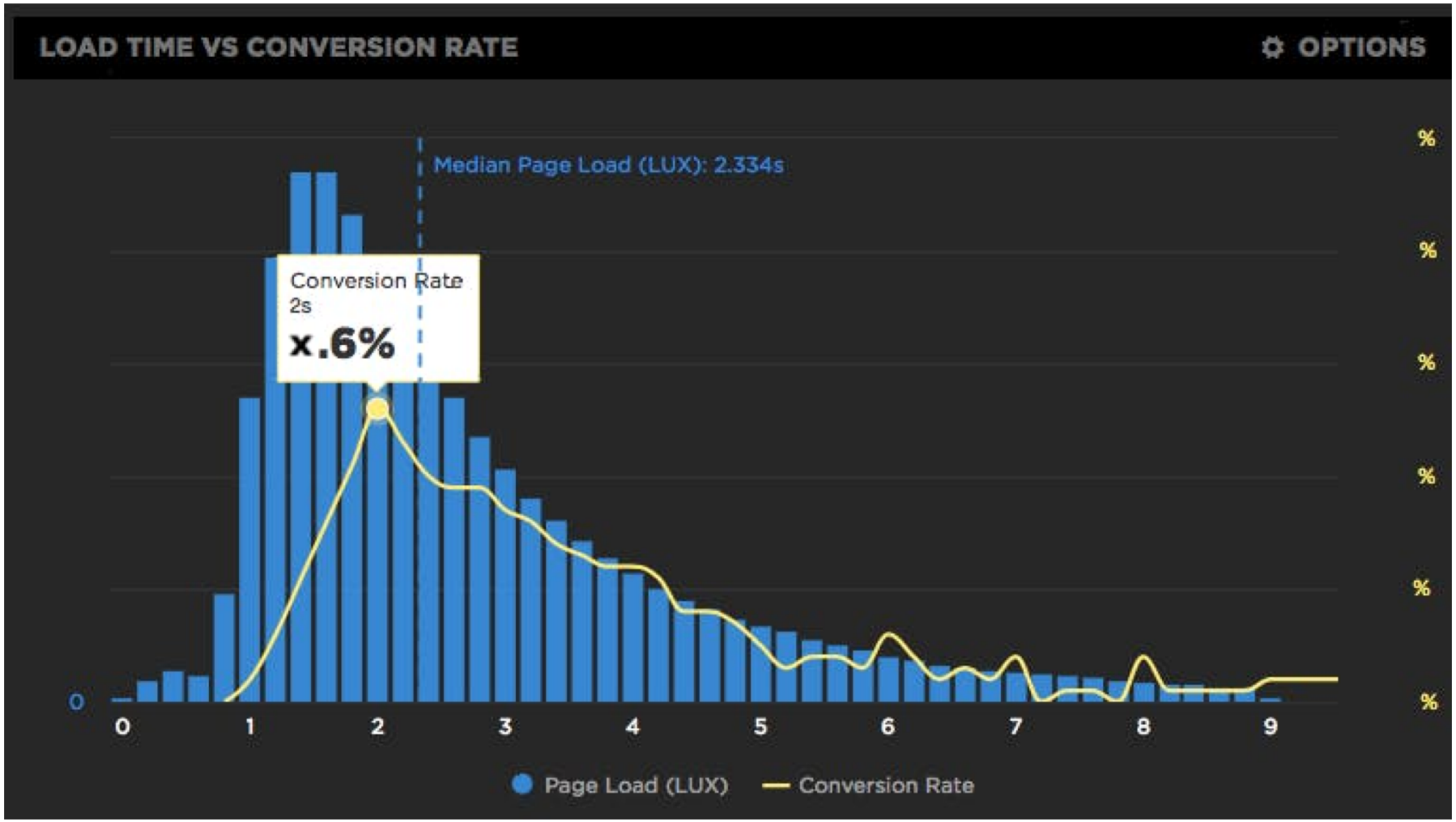
mPulse

Etc.

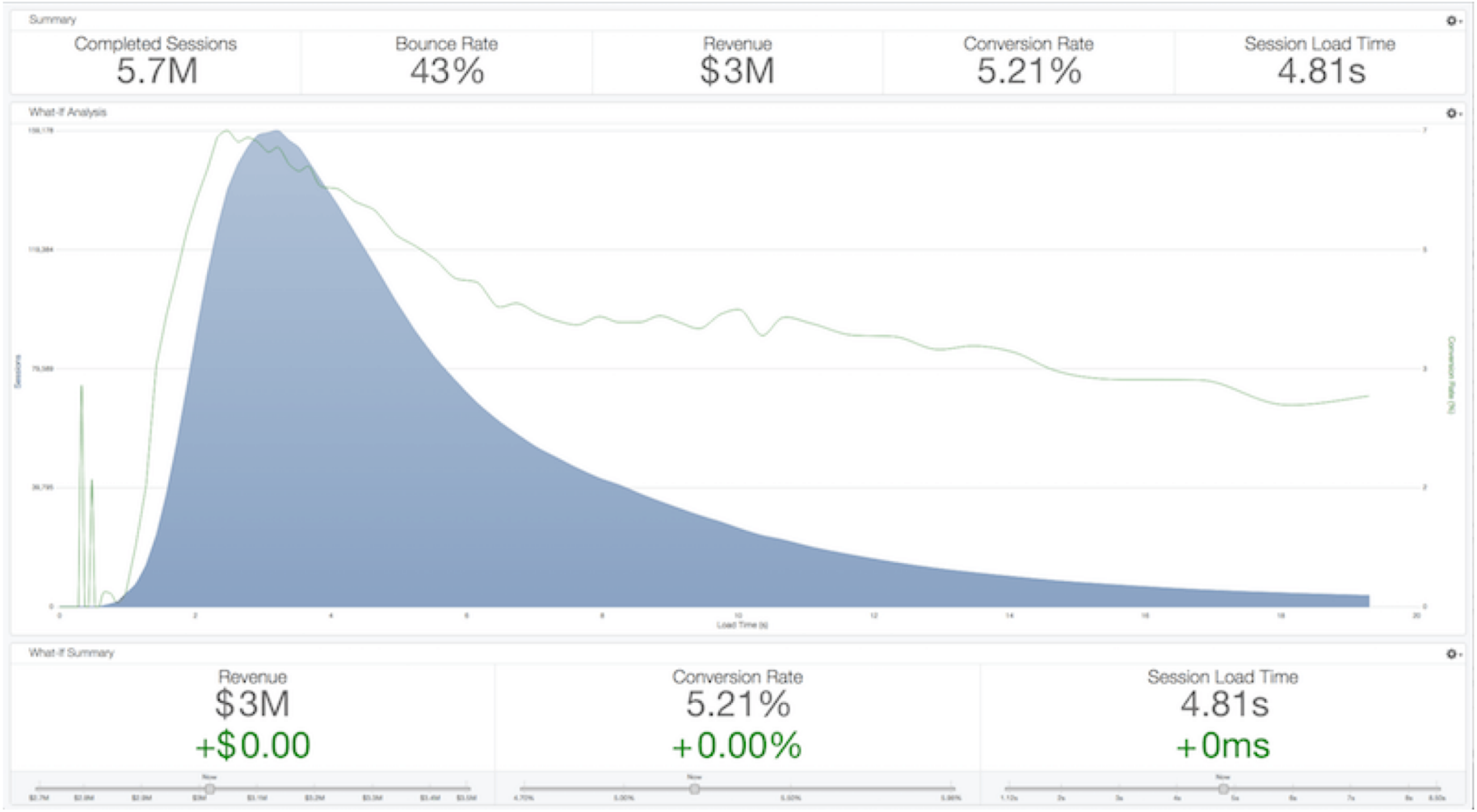
Google Analytics



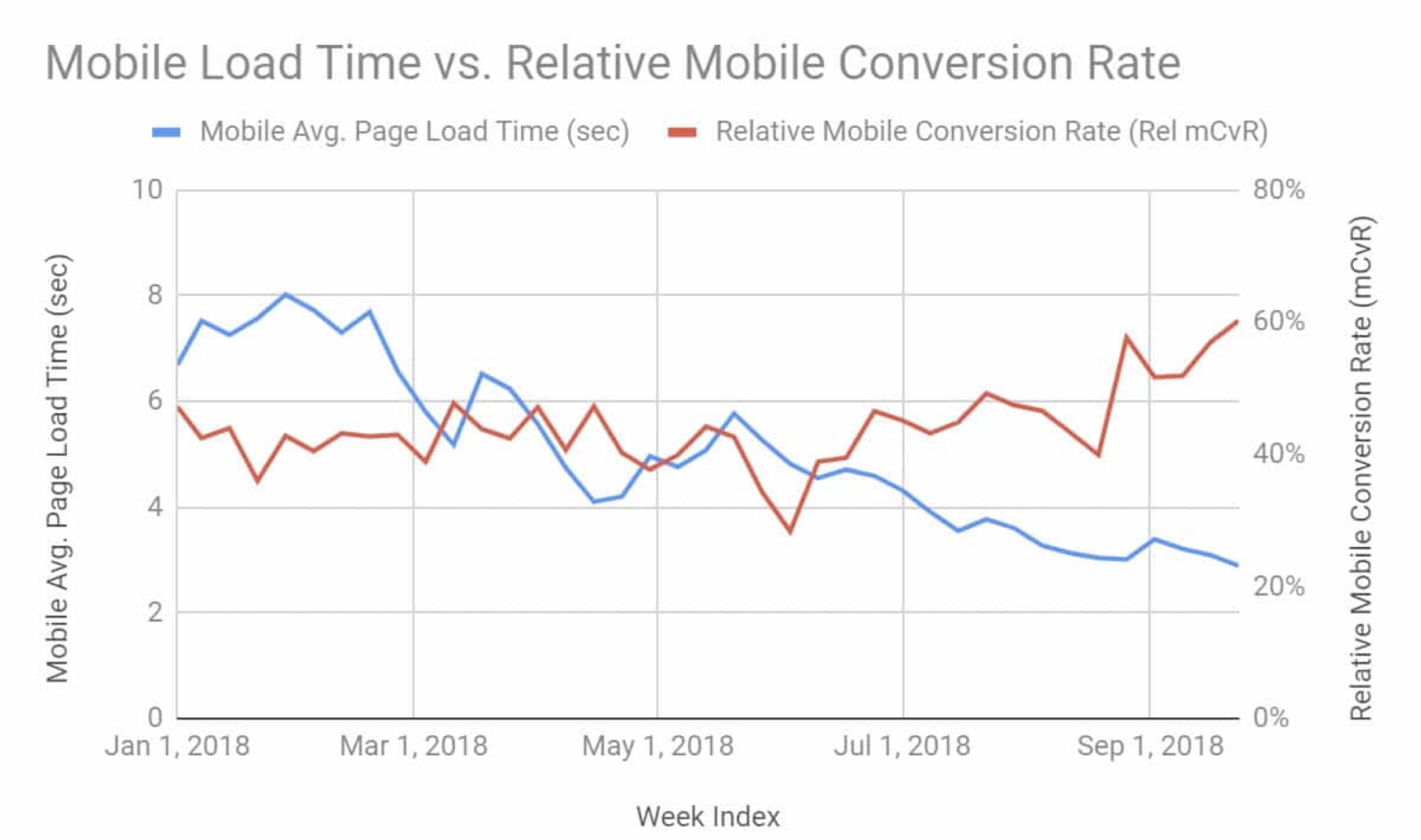
SpeedCurve


LOAD TIME VS CONVERSION RATE
OPTIONS
Median Page Load (LUX): 2.334s
Conversion Rate
2s
x.6%
%
%
%
%
%
%
0
0
1
2
3
4
5
6
7
8
9
Page Load (LUX)
Conversion Rate

Akamai mPulse

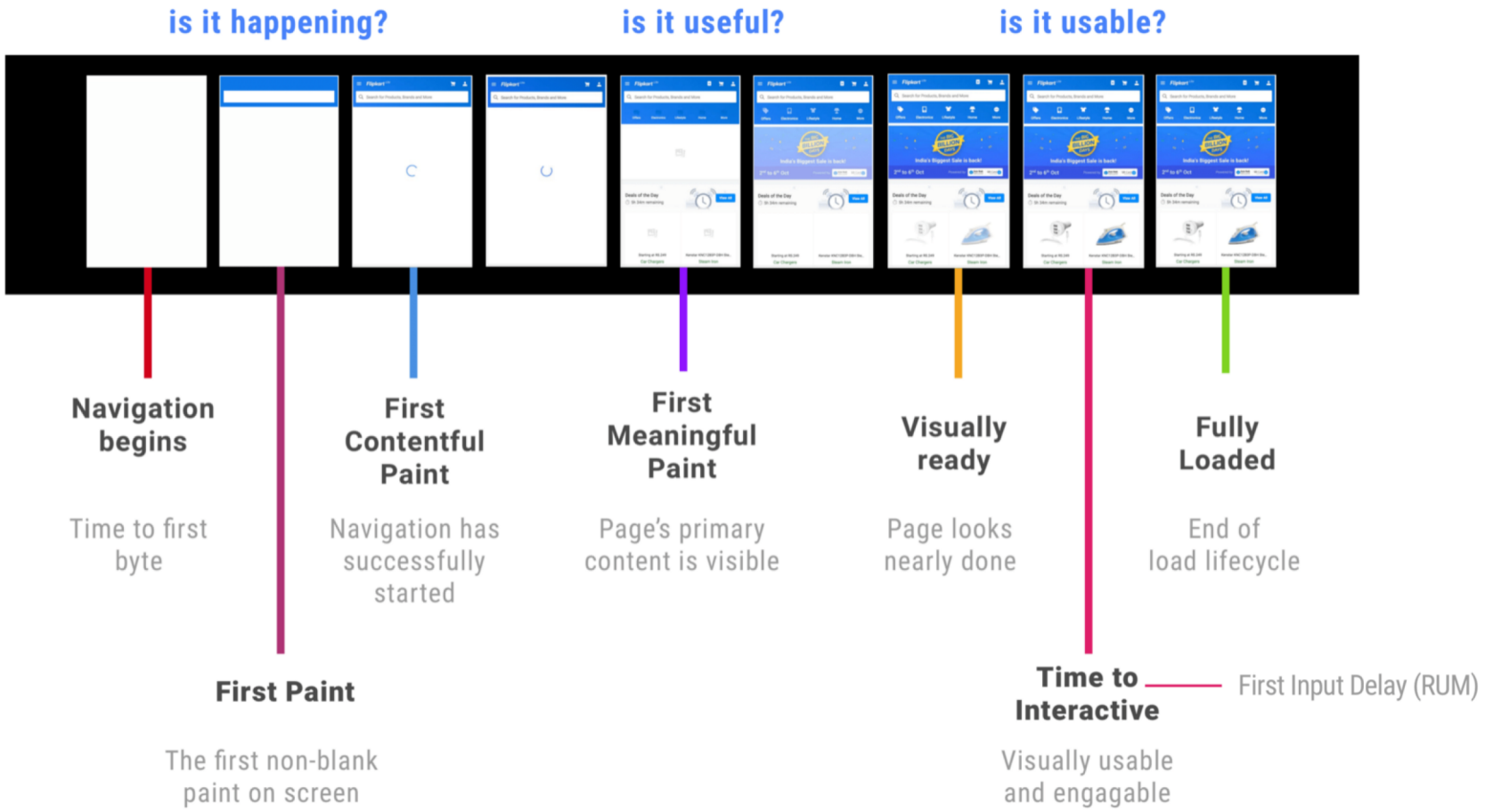


Corrélation entre performance et conversion



Source : https://web.dev/value-of-speed

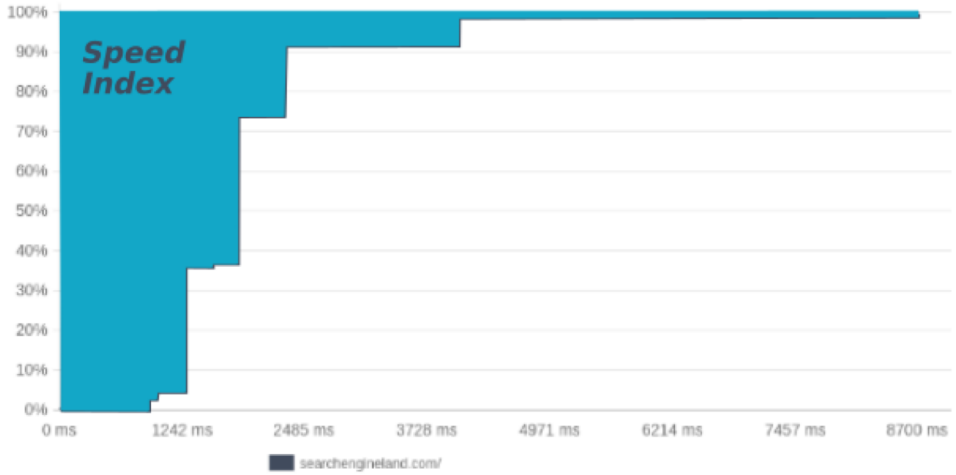
Différentes métriques à surveiller



Source : https://medium.com/@addyosmani/the-cost-of-javascript-in-2018-7d8950fbb5d4

Commencez par le Speed Index

Mesure de l'UX via la séquence de rendu du haut de page



Source : https://blog.dareboost.com/fr/2018/02/speed-index-performance-web/

2. Optimiser

Comment faire des sites performants ?

Des bonnes pratiques WebPerf connues **depuis plus de 10 ans**, toujours pas appliquées par la plupart des sites.

opquast
open quality standards

La requête la plus rapide est celle que l'on ne fait pas

Éviter les redirections

Ne pas faire charger des ressources non utilisées

- Polyfills JavaScript
- Images inutiles (en responsive notamment)

Utiliser les bons en-têtes HTTP de mise en cache

Limiter l'usage de services tiers

Réduire la latence et les temps de connexion

(DNS + connexion TCP + handshake TLS) × distance

Rapprocher la ressource de l'utilisateur avec des CDN

Limiter le nombre de domaines

Réduire le nombre de requêtes

- Bundles de CSS et JavaScript (mais pas trop)
- Sprites d'images (ou SVG inline)

Optimiser le poids des ressources

Minifier et compresser les assets textuels

- JavaScript et CSS systématiquement
- HTML, SVG éventuellement

Optimiser les images

- Utiliser les bonnes dimensions selon les mises en page
- Adopter les bons formats : WebP, JPEG-XR, etc.
- Limiter les méta données

Optimiser l'ordre de chargement des ressources

CSS et JavaScript critiques peuvent être inline

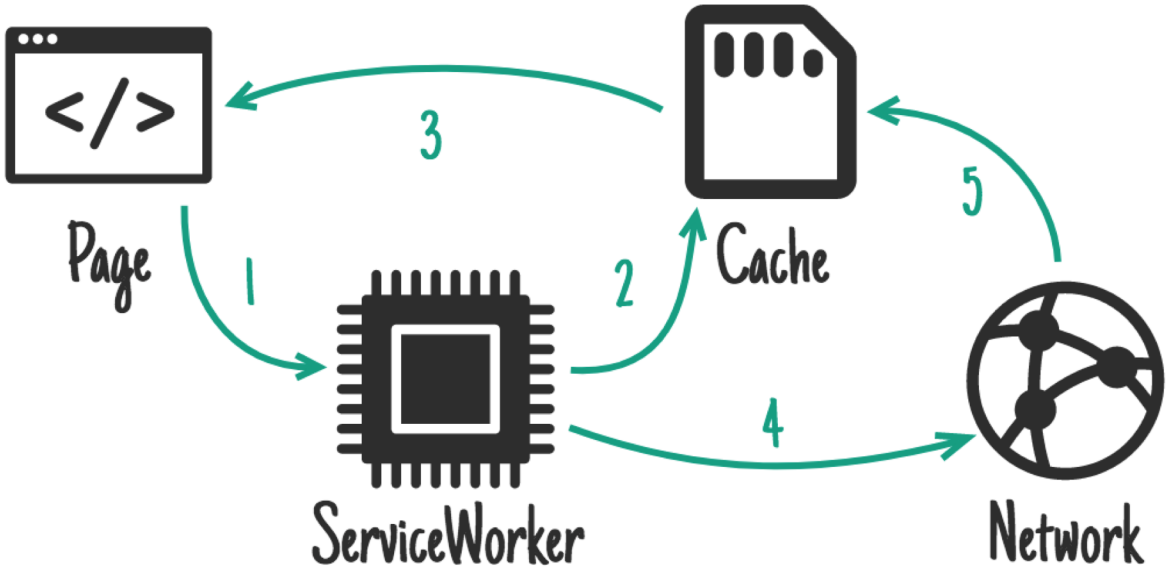
CSS et JavaScript non critiques peuvent être asynchrones

Seules les images visibles au dessus de la ligne de flottaison doivent être chargées rapidement

Profiter de la gestion des priorités de HTTP/2

Exploiter les Service Worker et l'API Cache

L'Offline First : facile techniquement, compliqué côté UX



Source : https://jakearchibald.com/2014/offline-cookbook/

3. Répéter

Comme le zéro défaut, la performance ultime n’est jamais atteinte…


WHERE WE'RE GOING
CLEVER AGE
WE NEED PERF